

REMARKS/ARGUMENTS

Amendments were made to the specification to correct errors and to clarify the specification. No new matter has been added by any of the amendments to the specification.

Claims 1-24 are pending in the present application. Claims 1, 4, 5, 10, 15, 18, 21, and 23 are amended. Claims 15 and 18 are amended into independent form including all of the limitations of the base claim to place claims 15-18 in condition for allowance. Support of the amendments to claims 1, 4, 5, 10, 21, and 23 is located at least on page 7, paragraphs [0091] through [0092] and on page 8, paragraph [0100] through page 9, paragraph [0109] of the patent application publication for the present invention. Reconsideration of the claims is respectfully requested.

I. Interview Summary

Applicants thank Examiner Susan Rayyan for the courtesies extended to Applicants' representatives during the December 19, 2006 telephone interview. During the interview, Applicants' representatives discussed the distinctions between the present invention and the cited *Chawla et al.* and *Mack* references. The substance of the telephone interview is included in the following remarks.

II. Allowable Subject Matter

The Office Action states that claims 15-18 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Applicants respectfully submit that the amendments to claims 15 and 18 place these claims in condition for allowance.

III. 35 U.S.C. § 103, Alleged Obviousness Based on *Chawla et al.* and *Mack*

The Examiner rejected claims 1-14 and 19-24 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication Number 2004/0181537 to *Chawla et al.* (hereinafter "*Chawla*") and U.S. Patent Number 6,801,915 to *Mack*. This rejection is respectfully traversed.

With regard to independent claims 1, 10, 21 and 23, the Examiner states:

As per independent claim 1 *Chawla* teaches:

detecting a response from a data store (paragraph 97-98, any result is retrieved); and placing the ... value in the response (paragraph 98, the result set is return to the end user client).

Chawla does not explicitly teach responsive to detecting the response, locating a merge reference section in a singleton in-memory object, identifying a merge reference from the merge reference section, determining a reference column from the merge reference and merging data with the reference column to form a merged value according to an order. *Mack* does teach this limitation (column 23, lines 34-67, merge reference

method and column values) to improve and maintain the quality of the data contained in the database. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Chawla with responsive to detecting the response, locating a merge reference section in a singleton in-memory object, identifying a merge reference from the merge reference section, determining a reference column from the merge reference and merging data with the reference column to form a merged value according to an order to improve and maintain the quality of the data contained in the database (column 3, lines 23-24).

...

As per independent claim 10, Chawla teaches a method in a data processing system for executing a request on a data store (Abstract) the method comprising: receiving a request containing data (paragraph 84,96).

Chawla does not explicitly teach ... locating a split reference section in a singleton in-memory object, identifying a split reference from the split reference section, determining a reference column from the split reference, extracting a value from the data and placing the value in the reference column according to an order. Mack does teach this limitation (column 24, lines 4-5, 21-30 and 47-57, as split reference data method) to improve and maintain the quality of the data contained in the database. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Chawla with ... locating a split reference section in a singleton in-memory object, identifying a split reference from the split reference section,

determining a reference column from the split reference, extracting a value from the data and placing the value in the reference column according to an order to improve and maintain the quality of the data contained in the database (column 3, lines 23-24).

...

Claim 21 is rejected based on the same rationale as claim 10.

...

Claim 23 is rejected based on the same rationale as claim 10.

Office Action dated October 3, 2006, pages 3, and 5-7.

As amended, claim 1, which is representative of the other rejected independent claim 21 with regard to similarly recited subject matter, reads as follows:

1. A method in a data processing system for generating return data responsive to a database request, the method comprising:
 - detecting a response from a data store after the database request specifying a requested field is executed, wherein a requesting client does not have knowledge of a database structure for the data store;
 - responsive to detecting the response, locating a merge reference section in a singleton in-memory object;
 - identifying a merge reference from the merge reference section;
 - determining a reference column from the merge reference;
 - merging the requested field with the reference column to form a merged value according to an order, wherein the merged value includes multiple columns from the data store in response to the database request specifying the requested field; and
 - placing the merged value in the response. (emphasis added)

As amended, claim 10, which is representative of the other rejected independent claim 23 with regard to similarly recited subject matter, reads as follows:

10. A method in a data processing system for executing a request on a data store, the method comprising:
- receiving the request from a client containing data, wherein the client does not have knowledge of a database structure for the data store;
 - responsive to receiving the request, locating a split reference section in a singleton in-memory object;
 - identifying a split reference from the split reference section;
 - determining a reference column from the split reference;
 - extracting a value from the data, wherein the data is split into multiple columns of the data store; and
 - placing the value in the reference column according to an order. (emphasis added)

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). For an invention to be *prima facie* obvious, the prior art must teach or suggest all claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

Chawla and *Mack*, taken individually or in combination, do not teach or suggest “merging the requested field with the reference column to form a merged value according to an order, wherein the merged value includes multiple columns from the data store in response to the database request specifying the requested field,” as recited in claims 1 and 21. Additionally, *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest “extracting a value from the data, wherein the data is split into multiple columns of the data store; and placing the value in the reference column according to an order,” as recited in claims 10 and 23. In addition, *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest that a client does not have knowledge of a database structure for the data store, as recited in claims 1, 10, 21, and 23.

Chawla is directed to a system with methodology for performing relational operations over relational data and data retrieved from Simple Object Access Protocol (SOAP) operations. In one embodiment, a method of the present invention is described for performing database operations on data obtained from a web service, the method comprising the steps of: creating at least one proxy table in a database, each proxy table mapping to a method of the web service; in response to a database operation on a particular proxy table, converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping; invoking the particular method of the web service; converting results obtained from invoking the particular method into data for use at the database based upon the corresponding mapping; and performing the database operation on the data at the database. As stated in the Office Action, *Chawla* does not teach the locating, identifying, determining, and merging steps of claims 1 and 21 and *Chawla* does not teach the locating, identifying, determining, extracting, and placing steps of claims 10 and 23. Specifically, *Chawla* does not teach or suggest

"merging the requested field with the reference column to form a merged value according to an order, wherein the merged value includes multiple columns from the data store in response to the database request specifying the requested field," as recited in claims 1 and 21, and *Chawla* does not teach or suggest "extracting a value from the data, wherein the data is split into multiple columns of the data store; and placing the value in the reference column according to an order," as recited in claims 10 and 23. In addition, *Chawla* does not teach or suggest that a client does not have knowledge of a database structure for the data store, as recited in claims 1, 10, 21, and 23. To the contrary, *Chawla* discloses that a user executes a SQL SELECT statement against a proxy table of the database server and therefore, the user/client has knowledge of the structure for the database.

Mack is directed to paired keys for data structures. A plurality of data records is provided, each data record having paired keys comprised of a first key and a second key. The first key identifies each data record in a certain manner and the second key identifies each data record in a different manner. The paired keys can be used, for example, by the first key to identify the original business identity which the data record belongs to, while the second key may identify the current business identity which the data record belongs to. In this way pre-merger data, for example, and post merger data can be retained. *Mack* does not teach or suggest "merging the requested field with the reference column to form a merged value according to an order, wherein the merged value includes multiple columns from the data store in response to the database request specifying the requested field," as recited in claims 1 and 21. Additionally, *Mack* does not teach or suggest "extracting a value from the data, wherein the data is split into multiple columns of the data store; and placing the value in the reference column according to an order," as recited in claims 10 and 23. In addition, *Mack* does not teach or suggest that a client does not have knowledge of a database structure for the data store, as recited in claims 1, 10, 21, and 23.

The Office Action refers to the follow portions of *Mack* in the rejection of independent claims 1, 10, 21, and 23:

The merge reference data method shown in FIG. 21 is used to combine two or more reference data records into a single reference data record. For this reference data method, we will assume that the new data record 2123 in FIG. 21 representing the active merged data record has already been created. This record is created using the create reference data method. The following steps are performed by a processor or a user to execute the merge reference data method:

1. Verify that the Company table's (such as table 2120), "Workflow Status" column (such as column 2144) values are "Active" for the data records, (such as records 2121 and 2122) that you selected to merge. If this is not the case, abort the data method. You can only merge reference data records if their "Workflow Status" column value is "Active".

2. Determine that no other reference data record is using the selected reference data record (such as records 2121 and 2122) to be updated by this the merge reference data method. The "Business Party ID" column's value (such as shown in Column 2142, "10" for record 2121 and "20" for record 2122) for the selected data record in the

company table (such as table 2120) needs to be unique in that column. If this value is not unique, abort the reference data method because some other data record is referencing the record you are attempting to modify. Reverse the offending reference data record(s) before trying the merge reference data method again. Update the "Workflow Status" value (such as in column 2144) to "Merge" for the reference data record in the company table you wish to mark as the merged reference data record. (Such as records 2121 and 2122). Update the "Business Party ID" column's value in column 2142 in the company table 2120 to equal the value of the active merged reference data record's (such as record 2123) "Business Party ID" column (such as column 2142, value "30"). Perform a database commit to store the changes in permanent memory.

Mack, column 23, lines 23-67.

The split reference data method (FIG. 22) is used to reverse the merge reference data method.

Mack, column 24, lines 4-5.

You can only split a reference data record if its "Workflow Status" column's (such as in column 2244) value is "Merge".

2. Update the "Workflow Status" column's value (such as in column 2244) to "Active" and update the "Business Party ID" column's value (in column 2242) to equal the value of the "Original Business Party ID" column 2243 for the reference data record (records 2221 and 2222) in the company table (such as table 2220) you wish to split from the previously merged reference data record.

Mack, column 24, lines 21-30.

Business party table 2310a includes data columns 2331a, 2332a, 2333a, 2334a, and 2335a. Person table 2320a includes data columns 2341a, 2342a, 2344a, 2344a, 2345a, and 2346a. Each of these columns now contains data. The data that is used to populate these tables is shown in FIG. 10A. The method used to populate these tables is shown in the flow chart of FIG. 25. Each data record is copied from source databases (FIG. 10A) into the new tables 2310 and 2320) using the create reference data record. After all the data records have been created, the merge and declare duplicate reference data method may then be executed to integrate and clean the reference data.

Mack, column 24, lines 47-57.

The merge reference data method of *Mack* is used to combine two or more reference data records into a single reference data record. To the contrary, claims 1 and 21 recite merging the requested field with the reference column. The split reference data method of *Mack* is used to reverse the merge reference data method. *Mack* does not teach or suggest "merging the requested field with the reference column to form a merged value according to an order, wherein the merged value includes multiple columns from the data store in response to the database request specifying the requested field," as recited in claims 1 and 21. Additionally, *Mack* does not teach or suggest "extracting a value from the data, wherein the data is split into multiple columns of the

data store; and placing the value in the reference column according to an order,” as recited in claims 10 and 23.

Chawla and *Mack* fail to teach or suggest “merging the requested field with the reference column to form a merged value according to an order, wherein the merged value includes multiple columns from the data store in response to the database request specifying the requested field,” as recited in claims 1 and 21. Additionally, *Chawla* and *Mack* fail to teach or suggest “extracting a value from the data, wherein the data is split into multiple columns of the data store; and placing the value in the reference column according to an order,” as recited in claims 10 and 23. In addition, *Chawla* and *Mack* do not teach or suggest that a client does not have knowledge of a database structure for the data store, as recited in claims 1, 10, 21, and 23. Therefore, the alleged combination of *Chawla* and *Mack* does not teach or suggest these features.

In view of the above, Applicants respectfully request withdrawal of the rejection of independent claims 1, 10, 21, and 23 under 35 U.S.C. § 103(a). Additionally, *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest the features of dependent claims 2-9, 11-14, 19-20, 22, and 24 at least by virtue of their dependency on independent claims 1, 10, 21, and 23, respectively. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 2-9, 11-14, 19-20, 22, and 24 under 35 U.S.C. § 103(a).

In addition to being dependent on their respective independent claims, claims 4-7, 13-14, and 20 are also distinguished over the *Chawla* and *Mack* references based on the specific features recited therein. Claims 4-7 are dependent on independent claim 1. *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest “identifying a merge reference from the merge reference section includes determining whether an identifier of the merge reference matches the requested field from a plurality of requested fields in the response,” as recited in claim 4. With respect to claim 5, *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest “merging the requested field with the reference column to form the merged value according to the order includes combining a value of the requested field with a value of the reference column.” With respect to claim 6, *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest “determining a reference column from the merge reference includes locating the reference column from a plurality of columns in the data store according to a value element of the merge reference.” With respect to claim 7, *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest that “the order is located in an order element of the merge reference.” To the contrary, the cited portions of *Mack* refer to consolidating data records.

Dependent claims 13, 14 and 20 are dependent on independent claim 10. *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest “identifying the split reference includes determining whether an identifier of the split reference matches a requested field from a plurality of

requested fields in the request,” as recited in claims 13. With respect to claim 14, *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest “determining the reference column includes locating the column from a plurality of columns in the data store according to a value element of the split reference.” With respect to claim 20, *Chawla* and *Mack*, taken individually or in combination, do not teach or suggest “determining a reference column further comprises determining a plurality of reference columns, and extracting the value further comprises extracting a plurality of values, wherein a first value of the plurality of values is placed in a first column of the plurality of columns, and a second value of the plurality of values is placed in a second column of the plurality of columns.” To the contrary, *Mack*’s split reference data method separates previously merged records of data.

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: January 3, 2007

Respectfully submitted,

/Gerald H. Glanzman /

Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

GHG/VJA